PATENT ABSTRACTS OF JAPAN

(11)Publication number : **63-225840**
(43)Date of publication of application : **20.09.1988**

(51)Int.Cl. **G06F 12/14**

(21)Application number : **63-050529**
(22)Date of filing : **03.03.1988**

(71)Applicant : **YOKOGAWA HEWLETT PACKARD LTD**
(72)Inventor : **ARAN DEI MAASHIYARU**
KURISUTOFUA JIEI MITSUCHIERU
GUREEMU JIEI PURAUDORA

(30)Priority
Priority number : **87 8704883** Priority date : **03.03.1987** Priority country : **GB**

(54) **INFORMATION STORAGE SYSTEM**

(57)Abstract:
PURPOSE: To simplify recalculation of an authentication code in a system which is only partially change normally, by calculating a general authentication code on the basis individual calculated authentication codes.
CONSTITUTION: Individual message authentication codes MAC of messages are calculated, and a global MAC of the whole of information which consists of plural files or messages and requires authentication is calculated on the basis of individual calculated MAC. These messages are divided into block each of which includes a considerable number of message for the purpor of making the system hierarchical. MAC of all messages in each block are calculated, and MAC of the block is calculated on the basis of MAC of all messages of the block. The global MAC is calculated on the basis of MAC of all blocks. Thus, MAC easily calculated in case of the change of only a part of information.

LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)特　許　公　報（Ｂ２）　(11)特許番号

第2675806号

(45)発行日　平成９年(1997)11月12日　(24)登録日　平成９年(1997)７月18日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　12/14	３２０		Ｇ０６Ｆ　12/14	３２０Ｂ	

請求項の数６(全 21 頁)

(21)出願番号　特願昭63－50529

(22)出願日　昭和63年(1988)３月３日

(65)公開番号　特開昭63－225840

(43)公開日　昭和63年(1988)９月20日

(31)優先権主張番号　８７０４８８３

(32)優先日　1987年３月３日

(33)優先権主張国　イギリス（ＧＢ）

(73)特許権者　999999999
ヒューレット・パッカード・カンパニー
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000

(72)発明者　アラン・デイ・マーシャル
イギリス国イングランド・ピーエス１・４アールジエイ・ブリストル・マーチヤンツ・ランデイング・トリン・ミルズ５

(72)発明者　クリストフア・ジエイ・ミツチエル
イギリス国イングランド・ピーエイ12ワン・ウイルトシヤ・ワーミンスタ・コツドフオード・ハイ・ストリート・メナ・ハウス・コツテイジ（番地なし）

(74)代理人　弁理士　上野　英夫

審査官　野仲　松男

最終頁に続く

(54)【発明の名称】　情報記憶システム

1

(57)【特許請求の範囲】

【請求項１】複数のメッセージから成る情報の本体を記憶する記憶手段と、

前記情報の本体を認証するメッセージ認証コードを計算する計算手段とを備えた情報記憶システムにおいて、

前記計算手段は、

前記複数のメッセージの各々から該メッセージの認証を可能にする各々の個別メッセージ認証コードを計算し、

前記個別メッセージ認証コードを１以上のメッセージとして扱い、該個別メッセージ認証コードの認証をするグローバル・メッセージ認証コードを計算し、

前記メッセージ認証コードのどれがどのメッセージに関係するか前記記憶手段に記憶する

ことを特徴とする情報記憶システム。

【請求項２】前記情報の本体を形成する前記メッセージ

2

から計算された前記個別メッセージ認証コードが、前記グローバル・メッセージ認証コードを計算する際に前記計算手段によって、ただ一つのメッセージとして扱われることを特徴とする請求項第１項に記載の情報記憶装置。

【請求項３】前記情報の本体を形成する前記メッセージから計算された前記個別メッセージ認証コードが、グループに分割され、個別メッセージ認証コードの各グループは、前記計算手段によってメッセージとして扱われ、グループメッセージ認証コードを計算するのに使用され、前記グループメッセージ認証コードは前記計算手段によって一つのメッセージとして扱われ、前記グローバル・メッセージ認証コードを計算することを特徴とする請求項第１項に記載の情報記憶装置。

【請求項４】前記情報の本体の前記メッセージの１以上

が変更された際に、前記変更された各メッセージに対する個別メッセージ認証コードと前記グローバル・メッセージ認証コードだけが前記計算手段によって再計算されることを特徴とする請求項第１項に記載の情報記憶装置。

【請求項５】前記情報の本体に１以上のメッセージを付加する際に、前記計算手段が前記付加された各メッセージの個別メッセージ認証コードを計算し、前記計算された新たな個別メッセージ認証コードと既存の個別メッセージ認証コードを使用して、前記グローバル・メッセージ認証コードを再計算することを特徴とする請求項第１項に記載の情報記憶装置。

【請求項６】前記情報の本体の前記メッセージの１以上が削除された際に、前記計算手段は、残された前記メッセージの前記個別メッセージ認証コードを使用して前記グローバル・メッセージ認証コードを再計算することを特徴とする請求項第１項に記載の情報記憶装置。

【発明の詳細な説明】

〔発明の技術分野〕

本発明は情報を安全に記憶する情報記憶方式に関する。

〔従来技術およびその問題点〕

コンピュータやデータ記憶システムにおいて、ユーザが情報を安全に（充分高いセキュリィをもって）、すなわち正当性を確認して、記憶することができることが必要とされる場合がある。これは情報の破壊に対しての耐久力があると言うことを意味してはいない。と言うのは部外者は記憶されている情報をほとんどいつでも破壊し得るからであり、情報を破壊から保護するには記憶手段の物理的な安全性（security）が必要とされる。ここで意味していることは、記憶されているデータが干渉を受けないということであり、このことはいかなる干渉も検出されるということを意味する。

実際には、これはメッセージ認証コード（message authentication code、MAC）によって達成される。MACを計算するには情報をMAC発生器に通し、これにより典型的には64ビット長のMACを得る。このMACを記憶しておくことができ、後にMACを再計算することにより情報を認証する（authenticate）することができる。もし記憶しておいたMACと計算で得られたMACが一致すればその情報は干渉を受けていない。もちろん、このMACはそれ自身干渉から保護されていなければならない、つまりこのMACを計算する本になっている情報の修正に釣り合うように部外者がはじめのMACを修正することから保護されていなければならない。この保護を実現するには、MACの計算に秘密のキーを用いる。MACを計算するに当たっての便利な、また以下で選ばれている一つの方法は、DES/DEA的なアルゴリズムとDES/DEA暗号化／解読ユニットを使用することにより、キーおよび暗号ブロックチェイニング（cipher block chaining、CBC）技術を用いる事である。このプロセスは情報を暗号化する場合と同じである。ただしMACを計算する際にはDES/DEAユニットからの出力ブロックのストリーム（暗号化された情報）は捨てられ、最後のブロックだけがMACとして保存される点が違っている。この技術を使用する場合は、MACそれ自身は情報と一緒に記憶され、MACを計算する為に用いられたキーだけが秘密に保たれる。

実際には、MACを安全な（正当性が確認された）情報記憶に用いるというこの技術は幾分面倒なものである。それはチェックしなければならない情報は非常に大量になりがちだからである。セッションの始めにユーザはチェックを始めるが、そのチェックでは記憶されている情報全体のMACの計算が行われる。セッションの終わりではユーザは情報全体についての新たなMACの計算をしなければならない。この情報はこのセッション中にその処理をすることによりユーザによって変更されているので、新しいMACはもちろん古いものとは異なってくる。

本願発明者の知見によれば、チェックされるべき情報は通常は多数の個別的なファイルまたは「メッセージ」からなっており、一回のセッションではユーザは通常その中の少数のものについてしか作業しない。従って、この情報のMACの計算にあたってはMACの変化に唯一寄与するところの変化したファイルに加えて、大量の変化していない情報、つまり変化していないファイル、をMAC発生器を用いてスキャンすることが行われる。しかしながら、変化したファイルだけからMACの変化を計算する簡単な方法はない。MACは連鎖状のプロセスである。先ず情報の64ビットの各々のブロックが順に先行するブロック群より計算されたMACと組み合わされてこの現ブロックまで（このブロックも含む）の全けのブロックについてのMACを得る。この情報の途中のあるブロックに変化によるMACへの影響を計算することはできない。

〔発明の目的〕

本発明の目的は上述した従来技術の問題点を解消し、情報の一部分しか変化しない場合のMACを簡単に計算することである。

〔発明の概要〕

本発明の一実施例によれば、複数のファイルまたはメッセージからなり、認証を必要とする情報の本体全体についてのグローバルMAC（全体的MAC）の計算が、各メッセージについての個別のMACを計算しこれら個別のMACからグローバルMACを計算することにより行われる。

最も単純な形では、メッセージの個別的なMACから直接的に計算され、これらの個別的MACは連結されてそれについてグローバルMACを計算すべき更に別のメッセージを実効的に形成すると見なされる。しかし、システムは階層化されていてよいということが理解できるだろう。このため、これらメッセージは夫々がかなりの数のメッセージを含むブロックに分割される。各ブロック毎にその全てのメッセージについてMACを計算し、そのブ

5

ロックの全てのメッセージのMACについてのブロックのMACを計算する。次にグローバルMACが全てのブロックのブロックMACについて計算される。従って、この場合でもグローバルMACはメッセージの個別のMACから計算され、間接的であるが、情報全体、すなわち全てのブロックの全てのメッセージ、正当性を確認する。ブロックMACはもちろん個々のブロックについて見ればそのグローバルMACになっており、個々にそのブロック中のメッセージの正当性を確認する。

記憶されている情報を何か改変すれば、それはMACの突き合わせの失敗を引き起こし、その改変が検出される。個々のメッセージの改変はそのMACの変化を引き起こす。MACは秘密に保たれているキーを用いて計算されるので、部外者は彼が改変したメッセージのMACを代えることはできず、そのメッセージのMACチェックは失敗する。もし部外者が余計なメッセージを挿入したり、完全なメッセージを取り除いたり、あるいはメッセージの順序を変えたりすると、グローバルMACの突き合わせの失敗が引き起こされる。記憶されている情報に加えられた改変の性質を判定することは通常は可能ではないが、このような改変の事実は常に明白になる。

この技術の利点は、もし実際のセッションでユーザが少数のメッセージだけを変更したのであれば、このセッションの終わりにおけるMACの計算では変化したメッセージのMACの計算とグローバルMACの計算が行われるだけである。この技術では、グローバルMACの計算はMACを一つだけ計算するものに比べるとオーバーヘッドであるが、しかしこのオーバーヘッドは比較的小さい。それは個々のメッセージのMACはメッセージそれ自体に比べて大幅に少ない情報しかないからである。メッセージが処理される（生成される、あるいは変更される）際にはいつもそのMACを計算しなければならないが、しかしどのセッションにおいても、そこで処理されたメッセージだけが再計算を必要とし、変化していないメッセージはそれにたいしていかなる計算も行われる必要がない。

本発明の補足的な側面は個々のメッセージを秘密を保って記憶することに関係するが、これはユーザがしばしば必要とするもう一つの特徴である。正当性の確認と同様に、これは情報が誤りに対して耐久力があるということを意味するものではない。ここで意味されていることは、記憶されている情報を部外者が読み出すことができないという保証があることである。

従って、本発明の実施例にはまた、安全モジュール内に一つあるいはもっと多くのキーを格納する手段と、暗号化／解読手段と、メッセージを記憶する前に暗号化する手段も設けられる。好ましくは、２以上の階層のキーが用いられ、ここで最下層のキーは各メッセージ用に夫々ランダムに生成されてメッセージ中に格納され、その階層構造のすぐ上位のキーと組み合わせられて暗号化キーをもたらし、階層構造が２よりも多い階層を有している場合には、その階層構造の上の方に向かって存在する各キーは最上位のキーを除いてはそのすぐ上にあるキーによって暗号化された上でメッセージの本体に夫々追加される。階層構造中の各キーは好ましくは予め定められた回数だけ使用された後に変更される。

6

メッセージは従って暗号化された形で格納され、ここで各メッセージはそのメッセージに固有のキーの下で格納される（なんとなれば、暗号化するためのキーはメッセージに固有のキーといくつかのメッセージに対して同一であるキーの組み合わせによって形成されるからである）。階層的キー構造を持つことおよび所与の量の使用の後にキーを変えることにより、部外者の暗号解読による攻撃の可能性が最小になる。

本発明は安全通信システムの分野に特に適用できる。パーソナルコンピュータのような端末が多数相互接続されている通信ネットワークはよく知られている。（以下の実施例では、セキュリティのため、暗号化キーを管理する端末であるキー分配センタ（key distribution centre、KDC）とユーザが使用する端末であるユーザ機器（user agent、UA）がネットワークに接続されている。）このようなシステムでは公衆電話システムのような安全でない、つまりセキュリティが充分でない通信媒体がしばしば用いられる。このようなあまり安全ではない通信媒体においては、受動的妨害（盗聴）や能動的妨害（メッセージを横取りして除去する、メッセージを改変する、あるいは不正なメッセージを挿入する）を受けやすい。これらの問題を克服するためには、暗号システムを設けることが知られている。しかしながら、暗号化の理論自体は明らかであるが、多数の端末を含むシステムを設計するに当たって係わってくる実際上の問題はかなりある。そのような問題の中に、情報を安全に記憶することに関連する問題がある。ここでの情報はユーザが生成したメッセージ（ユーザが生成してそのユーザの端末に格納されるメッセージと他のユーザによって受信されるメッセージの両者がある）およびシステムの構成上の目的のために用いられる情報の両方を指す。

本発明の更に別の局面は他の端末から受信したメッセージをこのような端末に秘密裡に記憶することに関連する。これもユーザがしばしば求める特徴である。

本発明の他の局面によれば、システムの遠方の端末から階層構造のキーの下で暗号化されたメッセージを受信してそのようなメッセージを格納する手段と、遠方の端末でそのメッセージの暗号化のために使用されたキーをキー階層構造の上の方へ向かって全て、但しその最上位のキーを除いてそのメッセージに追加する手段と、そのメッセージと付属部のMACを計算しこのMACをグローバルMACの計算に含める手段を設けた情報記録方式が与えられる。

〔発明の実施例〕

本発明の実施例の通信システムについて、図面を参照

して説明することにする。

説明は次の部分に分けて行う。

システムの全般的構成

システムの全般的動作―キーの階層

メッセージの構造とUAの構造

UAとKDCとの連鎖

各UA間の通話

システム・メッセージ・エラーの回復

ローカル・メッセージ記憶装置

UAの変更

KDCのメッセージの記録

本発明の他の特徴は本願と同時に提出した二つの同時係属特許出願に説明してあり特許請求されていることに注意すべきである。

システムの全体的構成

第１図を参照して、システムは、すべて共通の通信媒体11に接続されている複数の端末10、10A、10Bなどと、キーの制御と分配とに責任を持つKDC12とから構成されている。また非電子的物理的キー分配径路13があり、これによりキーはKDC12から端末10に分配されることができる。各端末10は、図示したとうりのパーソナル・コンピュータPC14やディスクメモリ15のような従来どうりの端末装置と、各種暗号キーである安全モジュール（security module）16とから構成されている。KDC12は安全モジュール17、計算ユニット18、および複数の記憶手段19から構成されているのでデータが失われる危険は無視できる。安全モジュール16と17は、二重の囲み線で示したように、外部の妨害に対して保護されている。

安全モジュール16は、制御目的で、PC14から制御線により信号も供給され、PC14への双方向データ径路を備えているように示してある。この後者の径路はデータをPC14から他の端末へ送る暗号化のため安全モジュール16に伝え、また他のモジュールからのデータの暗号を安全モジュール16で解説した後にPC14へ伝えるのに使用される。この径路は、端末内の局所的安全化（つまり暗号化された）ファイルのためおよびそのデータが再びアクセスされるとき解読するため、データをPC14から伝え、同じPC14に戻すのにも使用される。安全モジュール16はまた直接通信媒体11に接続されているように示してある。実際には、通信媒体11との或る形態のインターフェースが必要である。これは安全モジュール16でも行うことができるが、実際はこのためにはPC14で行うのが便利である。もちろん、これに関係するPCの部分は安全モジュールとの間で暗号化されていないデータのやり取りをする部分とは論理的に別である。（また、もちろん、PCは通常媒体11と直接交信して非安全メッセジを送受信する。）

安全モジュール16と17は既知の技術を使用して構成されている。したがって各モジュールは暗号キーや他の秘密の状態に保持しなければならない情報を格納するデータ記憶手段、データの暗号化や解読およびチェック用の数量の計算やモジュール内で必要なその他の処理のような動作を行う処理手段、および必要な動作を制御する制御手段を備えている。各モジュールは、万一一時的に局部的停電が起こった場合にキーのような安全情報が失われないように、電池も備えている。モジュールはまたモジュールに対する物理的攻撃を検知し、このような攻撃が起った場合にモジュールに格納されているすべての情報を、敵がモジュールを開き個人の構成要素に接続してモジュールから有用な情報を抜取ろうとする可能性に反撃を加えるように、たとえばキーのような格納情報にランダムなデータを重ね書きすることにより、破壊する手段を備えている。

各種レジスタ、カウンタ、および後に説明する記憶装置のような、モジュールの構成要素の多くは、好適にはマイクロプロセッサ、RAM、およびこれら構成要素に使用するメモリロケーションを規定したりその機能を実現したりする格納プログラム、により実現される。（ただし、乱数発生器や暗号化／解読ユニットのような或る構成要素は種々な理由から特殊目的のハードウェアで構成するのが都合がよい。）プログラムは少くとも一部はPROMまたは類似のものに格納されるので少くともプログラムの一部は一旦書き込まれてしまったら変更することはできない。したがってプログラムは格納キーまたは他の安全情報をモジュールから読み出すことができるように修正することはできない。

システムは通信媒体11に盗聴を行おうとする部外者20からの攻撃に対して開いているものと仮定している。このような部外者20はメッセージを傍受し、メッセージを横取りして取出し、本物のメッセージを修正して為のメッセージを挿入しようとする。通信媒体11は分散していて、端末10、10Aなどのいずれかのユーザの単独制御のもとにはない。たとえば通信媒体11は電話回線網または格納・先送り（store and forward）手段を備えたパケット・スイッチング・システムのような公共通話システムの一部を含んでいることがある。したがって部外者20の活動は本質的に検出できる性質のものではない。このような部外者による攻撃の可能性に加えて、通信媒体11は、メッセージが失われたり、その順序が変るようにメッセージに色々な遅れが加わったり、メッセージが重複（「エコー」）したりというような障害を本質的に受けやすいものと考えられている。

上記したこのような傍受の可能性や安全モジュールに対する物理的攻撃の可能性の他に、部外者は合法ユーザの留守中にモジュールにアクセスしてシステムに入ろうとする。これと戦うには、各種の技術を利用することができる。安全モジュールがパスワード制御を行うように設定し、パスワードを合法ユーザが入力して、モジュルはこうしてそのパスワードにだけ応答するようにできる。ユーザが彼の不在期間の長さを知っている場合に

は、時間ロックを使用するとができる。モジュールの内部電池によ時間ロックが確実に連続的に動作する。パスワードがモジュールにより生成され、これが合法ユーザが物理的に取外し且つ保持することができるフロッピーディスクに送られるようにすることもできる。

もちろん互いに多少異なる保護技術をユーザ端末の安全モジュールおよびKDCの安全モジュールに使用することができる。それはKDCは攻撃に対してユーザ端末より傷つきにくいようであるが、一方KDCに対する攻撃が成功すればユーザ端末に対するよりもはるかに阻害が大きくなるからである。

システムの全般動作－キーの階層

本システムの動作はKDC12により二つのレベルで制御される。第１に、各UAにKDCからユニークなユーザ・マスタ・キー（UMK）が割当てられる。このUMKは非電子式のキー分配径路13を伝わってUAに取込まれる。たとえばKDCを操作するスタッフの一員により、UA（の安全モジュール16内）に設置される。このキーはその後UAとKDCとの間のメッセージを確立したり確認したりするのに使用される。第２に、UAが他のUAと通話したい場合には、KDCを使用して二つのUA間に安全なチャネルを設定しなければならない。リンクを要求するUAは、KDCに通知し、KDCはこれにしたがってリンクを設定するが、その後リンクの使用には稀にしか（LMKが更新を必要とするとき）参加しない。第３のレベルのシステム動作も存在し、これは単独UAにおける情報の安全格納に関するものである。この動作はKDCには関係しない。

キーの物理的位置とその階層、および使用する略号を第１表に示す。各メッセージは、そのメッセージにどのレベルの階層が関連していようと、そのメッセージのためだけに発生された別々のメッセージ・キーで暗号化されるので、メッセージ・キーは階層になっているようには示してない。事実、各メッセージは一対のキーを使用して暗号化される。一つのキーは、基本キーと言うが、階層から取られるキーであり、もう一つはそのメッセージに対するメッセージ・キーである。

第１表

物理的位置

KDC－キー分配センタ

UA－ユーザ機器（user agent）（端末またはノード）

キー

UMK－ユーザ・マスタ・キー
（KDC　UA）

CKD－制御データ・キー　　MK－メッセージ・キー
（UA　UA）

LMK－リンク・マスタ・キー

LDK－リンク・データ・キー　　MK－メッセージ・キー
（UA内部）

PMK－パーソナル・マスタ・キー

PSMK－パーソナル・サブマスタ・キー

PDK－パーソナル・データ・キー　MK－メッセージ・キー

部外者が、同じキーで符号化した充分なメッセージを蓄積することができれば、彼は窮極的にシステムを破ってキーを取戻すことができる。したがってこの理由のためキーを適切な時間間隔で変更し、またそれ故部外者が何とかしてキーを手に入れたとしてもこのようなキーの更新の結果、それは結局彼の役に立たなくなるようにする。ただし、UMKは物理的に分散されているのでこれらを変更するのは困難である。したがってキーの階層システムを使用するのであり、このシステムでは、各キーは階層的にその上に位置するキーに変更がなされる前に繰返し変更される。上位のキーを使用して下位キーの変更に関する情報を伝達することができる。したがってKDCは比較的稀な時間間隔でキー（新しいUMK）の物理的輸送に関係するキー変更に関与することになるだけであり、このような更新は甚だしく厄介になることはない。

メッセージの構造とUAの構造

各種UAとKDCはメッセージによって互いに通信する。これらメッセージはすべてほとんど同じ構造をしているが、以下で示すとうり、変化がある。メッセージの基本的な分類の一つはシステム・メッセージとユーザ・メッセージとに分けることである。前者はユーザからは知ることなく、システムにより発生されたシステムにより操作するが、後者はユーザに応答して発生され、ユーザが組立てたデータを含んでいる。システム・メッセージは一般にかなり短く、幾つかの異なる形式がある。ユーザ・メッセージはその長さが非常に変動するが、実質上一つの形式しかない。わかるように、幾つかのシステム・メッセージを時々一つのパケットに組合せることができる。

メッセージの一般的構造、および始発UAでその発生に必要な、およびUAで受信した同様なメッセージに対する応答に必要なハードウェアについて、始めにUAとKDCとの間のメッセージを、特にこのような最初のメッセージを参照して、ここに説明することにする。他のシステム・メッセージはおおむね同じ方法で取扱われるが、小さな相異が、たとえば関連するキー・レベルに存在する。

システムが始動するのはUMK（ユーザ・マスタ・キー）がすべてのUA（ユーザ機器）に分配され設置されているが他のキーは存在せず通信キーも存在しない状態からである。第２図を参照すると、説明すべき各種要素を備えており、これら要素に対する一般的制御機能は制御回路30で行われている。UAに対するUMKはキー輸送ユニット31により物理的に輸送されるが、このユニットはUAの安全モジュール16に一時的に接続されてUMKをUMKレジスタ32に移す。使用カウンタ40はUMKレジスタ32と関連してUMKが使用された回数のカウントを保持する。このカウンタは、他のすべての使用カウンタと同様に、キーが使用されるごとにインクリメントし、最初（システムが最初に始動するとき）は０にセットされ、関連するキ

ーが更新される（すなわち新しいキーで置き換えられる）ごとに０にセットされる。

UMKは、比較的永続するが、充分使用してから、新しいUMKをKDCから物理的に輸送することによって更新することができる。そのためUMKキー番号レジスタ40Aが設けられており、これは最初０にセットされ、新しいUMKが設置されるごとにインクリメントする。（その代りに、UMKキー番号レジスタ40Aを、新しいUMKが設置され、KDCが新しい値を発生するごとにキー輸送ユニット31からセットすることができる。）

UMKがUAに設置されるとすぐ、制御データ・キーCDKが発生されてUAのCDKレジスタ34に格納される。ここで関連する使用カウンタ33とCDKキー番号レジスタ33Aも設定される。キーはランダム信号発生器36から発生される。ランダム信号発生器は熱雑音発生器または放射性崩壊カウンタのようなランダム信号源を利用してキーが確実にランダムになるようにしている。こうして発生したCDKは、適切なシステム・メッセージにより、即座にKDCに送られる。実際上は、このようなランダム信号発生器は比較的ゆっくりした速さでビットを発生し、したがって（典型的には64ビットの）次のキーのためのレジスタ（図示せず）を備えている。このレジスタの再補充はその内容が新しいキーのために取出されるとすぐ開始されるので、次のランダム・キーが即座に利用できる。このランダム信号発生器はしたがって、平文の（すなわち暗号で保護されていない）キーを持っているので、安全モジュールの中に入れられている。

メッセージを更に詳細に考察すると、UAはいくつかの区画を備えたメッセージ・アセンブリ・レジスタを備えており、ここでメッセージ（現在のものを含む）が組立てられる。メッセージ・アセンブリ・レジスタ37のMB領域はメッセージ本体あるいはデータ部であり、メッセージの「意味」または「データ」（もしあれば）を入れるのに使用される。メッセージ・アセンブリ・レジスタ37にはその左端に、ソース区画SCとデスティネーション区画DNの二つの部分がある。ソース区画SCにはこのUAを表わす不変のソースコードが格納されており、デスティネーション区画にはそこに送り込むのに必要なデスティネーションを示すコードが入る。次の区画MTは以下で説明するメッセージタイプ領域である。次の区画KNは、以下で説明するキー番号およびメッセージ識別子区画である。その次の区画はMK区画であって、これはメッセージのメッセージ・キーMKを入れるのに使用される。MB区画の次にはメッセージ認証コード（Message Authentication Code）区画MACが続き、前にはPMAC（以前のMAC）区画があって、これは当面無視する（または０が詰まっていると考える）ことがある。

メッセージタイプフォーマット記憶領域38は、たとえばCDKが送られているというKDCへのメッセージのような、システム・メッセージについての一組のメッセージタイプフォーマットを保持しており、この領域から適切なメッセージタイプが選択されてレジスタ37のメッセージ本体区画MBに送られる。KDCへのシステム・メッセージについては、この記憶領域はKDCデスティネーションコードをも保持する。もちろん、他のUAへのメッセージ（ユーザまたはシステム）については、受信UAのデスティネーションコードを発生しなければならない。他のUAとの通信はほとんどユーザが開始するので、受信UAのデスティネーションコードはユーザが決める。このコードはそのUAへのユーザ・メッセージによって使用される他に、もちろんそのUAへのシステム・メッセージによっても使用される。デスティネーションのニモーニックからデスティネーションコードを得る従来のテーブル探索システムをもちろん使用することができる。また、通信媒体11を通してのメッセージの径路決めあるいはアドレスシングは通常以下に説明するインターフェース・ユニット43により処理される。

上に注記したとうり、各メッセージは二つのキー、すなわちキー階層から生ずる基本キーとメッセージキー、を用いて暗号化される。メッセージ・アセンブリ・レジスタ37のキー番号区画KNにはメッセージに使用する基本キーを識別するとともにメッセージをユニークに識別するメッセージ番号としても働く、メッセージ用組合せキー番号が入っている。このメッセージ番号は、キーのキー番号を階層を下りながら基本キーに至るまで連結しまた基本キーの使用カウントも連結することによって得られる。各キー・レジスタは関連するキー番号記憶部、すなわちUMKレジスタ32についてはUMK用の40A、CDKレジスタ34についてはCDK用の33Aを備えている。したがって基本キーがUMKであれば、UMKキー番号レジスタ40Aおよび使用カウンタ40の内容を使用し、基本キーがCDKであれば、CMKキー番号レジスタ40A、CDKキー番号レジスタ33A、および使用カウンタ33の内容を使用うする。各キー番号は関連キーが変るごとに１だけ増加する。したがって所与のメッセージタイプに対して、メッセージ番号は厳密に昇順である。何故なら各キーのキー番号は通常上昇し、このような番号が０にリセットされることにより下降するときは、上位の番号の増加の結果だからである。関連する階層の分岐および階層を下る距離はメッセージタイプからわかる。たとえば、ここで考えているメッセージタイプ「CDKがKDCに送られている」については、キー階層は必然的にUMKだけしか含んでいない。

キーのキー番号は階層中ですぐ上のキーの使用カウントと類似しているが、この二つは必ずしも同一ではないことに注意すべきである。これは或る状況では階層内の高い方のキーを使用することができ、したがってその使用カウントが、階層中で直下のキーを変更せずに、増加するからである。このような問題を避けるには使用カウントとキー番号とをシステムを通じて別個に維持する。（このことはまたメッセージ番号は必ずしも連続ではな

いことを意味している。）

（このシステムはメッセージタイプを平文で示すMTの内容にある程度依存している。これはもちろん、たとえばMTの内容を暗号化されるものの一部に含めることにより修正することができる。この場合、メッセージ識別子（すなわちキー番号）の長さ、あるいは、これに相当する、階層内のキーのレベルは別々に示さなければならない。）

一般に、各メッセージの本体はそのメッセージにユニークなキー、メッセージ・キーMK、を使用して暗号化される。このメッセージ・キーはランダム信号発生器RND36を用いてUAにより発生され、メッセージ・キー・レジスタMK39に送られる。

使用する暗号システムは、DES/DEA規格または同様もののような、暗号化および解読に同じキーを使用するものであると仮定する。（「パブリック・キー」システムのような、暗号化と解読とに異なるキーを用いるシステムを使用することは可能であるが、キー対の両方のキーを格納し、また暗号化および解読のために適切な方を使用する必要がある。）使用する暗号化技術はCBC（Cipher Block Chaining）であり、これには初期設定ベクトルIVと暗号キーが必要である。（これについてはたとえばANSI規格X3.106－1983DEAの動作モードに述べられている。）初期設定ベクトルIVは最初基本キーのもとでメッセージ・キーMKを暗号化することにより作られる。次にメッセージの暗号キーは基本キーのもとで再びIVを暗号化することにより得られる。メッセージ・キーMKは平文で送られ、受信側は基本キーのコピーを備えているので、メッセージは、メッセージ・キーを基本キーのもとで暗号化して初期設定ベクトルを得、再び解読キーを得ることによって、他端では解読することができる。IVと解読キーは次にメッセージを解読するのに使用される。各メッセージに異なるMKを使用することは、メッセージがほとんど同じ形（たとえば同じユーザ・メッセージが２回目にはおそらく時間の違いだけで送られる）で繰返されても異なるキーのもとに暗号化されることになり、部外者は暗号を侵害しようとするに際し繰返しから多くの援助を得ることができないことを意味する。

メッセージが暗号化されたら、メッセージ・アセンブリ・レジスタ39のMT、KN、MK、PMAC、およびMBの各部の内容がメッセージ認証コード計算ユニット42に送られ、ここでMAC値が計算され、この値がメッセージ・アセンブリ・レジスタ37のMAC区画に送り返される。このようにしてMAC値がメッセージの一部として含まれる。MACはCBC技術を用いて暗号化類似プロセス（「MAC暗号化」）により計算される。このプロセスから得られる最終ブロックがMACを形成する。「MAC暗号化」はキーと初期設定ベクトル（IV）を用いて行われる。キー（「MAC暗号キー」）はメッセージを暗号化するのに使用する基本キーの固定された関数として得られ、IVは０とされる。ソースコードおよびデスティネーションコードは認証される必要が無い。何故なら、どちらかがどうにかして変化すれば、実際にメッセージを受けるユニット（UAまたはKDC）が、MACチェックにより認証を行なおうとする際適格なキーを使用していないので、メッセージを認証することができないからである。

暗号化／解読ユニット41とメッセージ認証コード計算ユニット42はキーを平文で受取らなければならないから、安全モジュールの内部になければならない。同様に、UMKレジスタ32は、キーを平文で持っているので、これも安全モジュールの内部になければならない。モジュールの外部に他のキーを格納し、UMKのもとで暗号化し、必要なときモジュール内で解読することも可能であるが、すべてのキーをモジュール内のレジスタに平文で格納するのがはるかに便利である。メッセージ・アセンブリ・レジスタ37ももちろんモジュールの内部にあり、暗号化され認証される前にメッセージが侵されないようにしておく。

送受される各メッセージは、通信媒体11と結合するのに必要となる低レベルのプロトコル処理を行うインターフェース・ユニット43を通過する。特に、インターフェース・ユニット43は暗号メッセージの伝達に専念するメールボックス、あるいは、このような、一つはシステム・メッセージ用の、もう一つはユーザ・メッセージ用の、二つのメールボックスとすることができる。これにより更に別のメールボックスを暗号化されないメッセージのために使用することができ、これら暗号化されないメッセージは平文で（たとえば安全通信システムの一部分を形成しない端末から）送受信される。上に注記したように、このインターフェース・ユニットはPC14（第１図）により好都合に実現される。

今度は受信回路を考察すると、通信媒体11からインターフェース・ユニット43を経て受信される着信メッセージを受取るのにメッセージ・アセンブリ・レジスタ37が使用される。このメッセージのメッセージ番号KNは対応する基本キーが利用できることをチェックするために調べられる。次にメッセージのMACが、メッセージ識別子KNにより「MAC暗号化」キーとして識別される基本キーを使用して、メッセージ認証コード発生器ユニット42によりチェックされる。得られたMACはコンパレータ44によりメッセージの（MAC区画にある）MACと比較される。MACの計算値とメッセージのMACとが合致すれば、メッセージは本物と判定される。合致しなければ、どちらかに（おそらくは送信雑音の結果）誤りが存在するかあるいは改竄されているので、メッセージは捨てられる。部外者20がメッセージを修正しようとしても、部外者には未知のキーを用いて計算することにより保護されているメッセージのMACを彼は訂正することができないので、変更されたメッセージと一貫していなければならないメッセージのMACを変更することはできないであろう。

15

次にメッセージのメッセージ番号KNはメッセージが前に受信されたものの繰返しではないことをチェックするために調べられる。メッセージ番号が合理的で前に受取ったメッセージと関係があるかどうかを知るためにチェックすることができる。（失なわれたメッセージ、重複メッセージ、および受取り順序が不良のメッセージに関する備えについては後に詳細に説明する。）

メッセージがKN試験およびMAC試験を通過すれば、（メッセージ本体部分MBが空でないと仮定して）メッセージは解読される。このため、MK区画のメッセージ・キーが（メッセージ番号により識別される）基本キーのもとに暗号化／解読ユニット41を用いて暗号化されてIVを得、これが再び基本キーのもとで暗号化されて解読キーを得る。（解読用のIVおよび解読キーは暗号化用のIVおよび暗号キーと同じである。）IVおよび解読キーは直接暗号化／解読ユニット41に送られてMB区画の内容を解読するのに使用される（或るシステム・メッセージ、たとえば或る承認メッセージは「本体」を備えていない。そのMB区画は空である）。

それでMB区画の内容は或る種のシステム・メッセージであり、これは制御回路30により処理される。なおこのメッセージがKDCから受信されているものとすると、メッセジはCDKキーを備えてよい。もしそうなら、その受信したキーはCDK1レジスタ46またはCDK2レジスタ47に送られる。KDCのCDKが変化した場合でも以前のCDKを使用しているメッセージが新しいCDKが受信された後でも受信されることがあるため、一対の受信CDKレジスタが存在する。二つの受信CDK番号レジスタがあり、これらには（やはりMB区画から）対応するCDK番号が送り込まれているので、CDKで暗号化されたメッセージを受信したとき適切なCDKを識別することができる。CDKはそのもとで暗号化された固定されたかなりの数のメッセージが送出されてはじめて変えられるから、以前のCDKを二つ以上保存しておくことは決して必要が無いと仮定しても危険ではない。（捨てられたCDKが必要になった場合には何か他に根本的に悪いものがあるであろう。）

UAとKDCとのリンク

システムは、すべてのUA（ユーザ機器）にUMK（ユーザ・マスタ・キー）が設置されているが他にはキーが存在せず通信リンクも存在しない状態で、始動する。UMKがUAに設置されるとすぐ、UAとKDCとの間にリンクが設置されなければならない。これを始めるには、制御データ・キーCDKを発生してUAのCDKレジスタ34に格納し、システム・メッセージを（レジスタ39にあるそのユニークなメッセージ・キーMKを用いて）UMKのもとで暗号化して構成し、KDCに送信する。こうしてKDCはUAがUMKを設置したこと、およびCDKがUAとKDCとの間のリンクの両端に設置されたことを知るので、CDKをUAからKDCへの将来の通信に使用することができる。KDCは承認メッセージをUAに送り返してCDKを受取ったことを認める。

16

その他に、KDCは、同じ方法で、このUA用にKDC自身のCDKを発生し、UMKのもとで暗号化して、UAに送信する。UAはこのメッセージを受信し、これを解読してKDCからのCDKを得る。こうしてUAとKDCとの間に、各方向に一つづつの一対のCDKを用いてリンクが設置される。リンクの両端は今後のメッセージを暗号化するために発生されたCDKを使用するとともに、リンクの他端から受信する今後のメッセージを解読するため他端から受信したCDKを使用する。メッセージを送ることができる二つの方向に対してこのように一対のキーを使用することはUA同志の間のリンクの場合にも行われる。

UAからのCDKを備えたメッセージをKDCが受取ったことの承認を別々の異なったメッセージにする必要はないが、その代りKDCからそのCDKをUAに送信するメッセージの一部として入れることができる。そのメッセージは今度はUAにより承認される。したがってCDKの交換は三つのメッセージで行われる。すなわち、UAからKDCへのCDKと、KDCからUAへのCDKによる受信の承認と、UAかKDCへの承認とである。

このようにUAとKDCとの間のリンクは各方向に別個のCDKを備えた双方向のものである。この方法でUAとKDCとの間に一旦リンクが設置されると、両ユニット間の今後のほとんどすべてのメッセージはCDKを基本キーとして用いる。CDKの使用が所定限度を超えると、新しいCDKが作られ、上述のようにUMKによる暗号のもとに送信される。UAとKDCとの間のメッセージの流れは比較的少ないので、この２レベル（UMKとCDK）の階層は、UMKの変更を稀にしか必要としないシステムき当分の間動作させるのに充分である。事実、UMKの使用は、以下でわかるように、新しいLMKが必要なとき、UA間の或る通信にも依存する。UAとKDCとの間のメッセージは一般にユーザ（UA）が他のユーザ（UA）とのリンクを設置または破壊したいときにのみ必要であり、これは（リンクは実質的に永続的であると見なされているので）稀に、しかもLMKを更新する場合にしか起らない。

一般に、リンクを伝わって二つの方向の流れるメッセージ（ユーザであろうとシステムであろうと）の数は互いに同じである必要はない。したがって続いて起る新しいCDKの個々の更新ではリンクのCDKの一方の更新しか行なわれない。このような更新では新しいCDKを或る方向へ送出し、これに対する承認メッセージが逆方向に送出される。

必要ならば、システムが最初に立上げられる際必要なすべてのリンクを設定するように、KDCをプログラムすることができる。これを行なうには各UAのキー輸送ユニット31（第２図）にかなりの数のシステム・メッセージを格納する。これからシステム・メッセージは暗号化が不必要である（このメッセージはUMKと共に輸送され、その安全性は、UMKの安全性と同様に、物理的であるため）。もしこうしなかったなら、これらのシステム・メ

17

ッセージは、システムが最初に動作状態になったときKDCと各UAとの間で送信されなければならないことになるものである。これによりKDCに関するシステム・メッセージの最初の数が著しく減少する。KDCの構造はUAの構造と同じであり、第３図にブロックの形で示してある。制御ユニット50（第２図に示すUAの制御ユニット30に対応）と一つのメッセージ・アセンブリ処理回路51が設けられ、メッセージ・アセンブリ処理回路51にはメッセージ・アセンブリ・レジスタ52（メッセージ・アセンブリ・レジスタ37に対応）がある。暗号化／解読ユニット、メッセージ認証コード計算ユニット、およびMACコンパレータの関連回路はここではメッセージ・アセンブリ処理回路51の一部と見なしてあり別個には図示してない。KDCには各UAに対してキー・レジスタと使用カウンタの集合体が個別に設けられている。ここでは、送出の場合KDCが使用するキー（すなわちレジスタ32、34、および39、および関連する使用カウンタおよびキー番号レジスタに対応する）、およびUAがKDCにメッセージを送る場合に使用するキー（すなわちレジスタ46と47および関連レジスタ48と49に対応する）の各集合体を、ブロック53、54、55、……で示してある。ブロック53、54、55、……はセレクタ回路61で制御されるマルチプレクサ60によりメッセージ・アセンブリ処理回路51に対して多重化されている。セレクタ回路61の内容により、ブロック53、54、55……から適切な一つを選択して受信メッセージを処理し送出すべきメッセージを準備するキーを得る。このようにしてメッセージが受信されると、セレクタ回路61にメッセージ・アセンブリ・レジスタ52のSC区画の内容が入れられる。この区画に受信メッセージのソースコードが入っており、従ってどのUAからメッセージが来たかを識別する。受信メッセージを送出したUAに応答メッセージを送り返さなければならない場合には、セレクタの内容は変更されず、これによりブロック53、54、55……のうちの適切な一つが応答メッセージの準備のため選択されたままになる。しかし、メッセージを別のUAに送出しなければならない場合には、セレクタ回路61の内容をもちろんそれに従って変えなければならない。これは、たとえば、リンクを設置している最中に生ずる。UA1からKDCへのリンクの設置を要求するメッセージには、そのMB区画に、UA2向けのコードが含まれており、このコードは適切なメッセージをUA2に送出するためセレクタ回路61に転送しなければならない。二つのコードはこの場合、メッセージがUA1およびUA2へまたこれらから送受されるので、セレクタ回路61により交互に使用される。

UA間の通信

通信が可能なためには、UA対の間にリンクが設定されなければならない。このような各リンクはUAがKDCに自分と他の指定したUAとの間にリンクの設定を要請することによって設定される。KDCは、リンクに含まれること

18

になるいずれのUAも他のUAとの間で備えることができるリンクの数に関する上限を超えていず、要求されたリンクの「受信」端にあるUAがリンクの受入れを拒絶しなければ、要求されたリンクを設定する。一旦リンクが組立てられてしまうと、両端のUAは同等の立場に立っているという点で対称である。どちらも他に対して送信することができ、あるいはリンクを切る決断をすることができる。

リンクを設定するプロセスを第IIA表に要約してあるが、この表ではリンクを要求するUAはUA1と呼ばれ、UA1がリンクを持ちたい相手のUAをUA2と呼んでいる。

第IIA表

UA1→KDC:UA1がKDCにUA2とのリンクを要請する。
UA2←KDC:KDCがUA2に送信LMKと受信LMKを送る。
UA2→KDC:UA2が受信を確認する。
UA1←KDC:KDCがキーをUA1に送る。

一層詳細には、UA1のユーザが、UA1のユーザにより指定された他のUAであるUA2とリンクを設定したいとき、UA1はシステム・メッセージをKDCに送る。このシステム・メッセージは、UA1がKDCへのシステム・メッセージに使用する（もちろんCDK更新に関連するシステム・メッセージを除く）キーである、UA1のCDKキーを基本キーとして使用し暗号化して送られる。そのメッセージ・タイプはUA1がリンクを設定することを要求していることを示し、メッセージ本体はUA2のコードを含んでいる。このメッセージを受信すると、KDCは一対のランダムはLMKを作り、メッセージをUA2に送る。そのメッセージのメッセージタイプはUA2にUA1とのリンクを受入れたいか否かを尋ね、またメッセージ本体はUA1および二つのLMKのコードを有している。これらはすべて基本キーとしてKDCからUA2にメッセージを送るのに使用されるCDKを使用して暗号化されている。UA2がこのメッセージを受取ると、そのユーザはリンクを受入れるか否かの意志決定をしなければならない。リンクが受入れられれば、メッセージはUA2からKDCに送られる。このメッセージはリンクの受入れを示しまたUA1のコードを含んでいる（UA1のコードをここに入れるのは他のUAに関連する設定用メッセージから区別するためである）。このメッセジはまた、メッセージをUA2からKDCに送るのに使用されるCDKを基本キーとして使用して暗号化される。KDCは、このメッセージを受取ると、メッセージをUA1に送ってUA2によるリンクの受入れを示すとともに、メッセジをKDCからUA1に送るのに使用されるCDKを基本キーとして使用して暗号化したUA2のコードおよび二つのLMKを取入れる。この結果、二つのUAすなわちUA1およびUA2は今は互いに直接通信するのに使用することができる一組のLMKを共有することになる。

リンクを設立できない一定の状況が存在する。実際問題として、UAにはこのようなリンクを維持するための限られた容量しか設けられていない。したがってUA1が既

19

に可能最大数リンクを持っている場合には、他のリンクを設定しようとすることを拒むことになる。ユーザには現存するリンクを切ってそのUAが新しいリンクを受入れる容量を作り出すようにする選択権がある。また、UA2が既に可能最大数のリンクを持っていることもある。そのときはKDCにシステム・メッセージを戻してこの旨を示し、KDCは今度はシステム・メッセージをUA1に送って要請したリンクが拒絶されたことを示す。（望むならば、UA2をそのユーザにUA1がリンクを要求していることを示し、そのユーザが現存するリンクを切って、要求されたUA1とのリンクを受入れる容量を作り出すように構成することができる。）加えて、上に記したように、UA2にこのような能力があれば、そのユーザは要求されたリンクを受入れるべきか否かを尋ねられ、もしユーザが拒絶すれば、UA2は再びKDCにこのことを示すシステム・メッセージを送る。このようなシステム・メッセジをKDCに送るとKDCは何が起ったかを示す対応メッセージをUA1に送り、UA1のユーザは要求したリンクが拒絶されたことを知る。（安全システムではユーザの要求が拒絶されたとき、拒絶の理由が示されないのが普通である。）

第４図に、第２図に示したよりも概略的にUAの構成を示す。メッセージ・アセンブリ処理回路はブロック５で示してあり、メッセージ・アセンブリ・レジスタ37、暗号化／解読ユニット41、およびメッセージ認証コード計算ユニット42を備えている。数ブロックのキー・レジスタおよび関連回路が存在する。ブロック70は第２図に示す各種キー・レジスタとそれに関連するカウンタとを含んでおり、すべてKDCとの通信に関係する。ブロック71、72、……は同様なキー・レジスタとカウンタを備えているが、各ブロックは別々のUAとの通信と関係する。従って、これら各ブロックはどのUAがそのブロックに関連しているかを識別するUAアドレス・コード・レジスタ（レジスタ73）を備えている。これらレジスタには、当該UAのユーザが他のUAとのリンクを要求して認可されたとき、および他のUAが当該UAとのリンクを要求し認可されたとき、この他のUAのアドレス・コードが入れられる。ブロック70、71、72,……はマルチプレクサ74により選択される。KDC用のブロック70の場合、選択はメッセージ・アセンブリ・レジスタ37のSC区画または制御回路30により直接制御される。他のブロックの場合には、選択は（着信メッセージに応答して）メッセージ・アセンブリ・レジスタ37のSC区画にあるアドレス・コードと各種レジスタ73の内容とを比較することにより決定される。送信メッセージの選択の場合には、選択はユーザが決定する（実際にはそのアドレス・コードに対してユーザが定義したUA識別子を格納するPC14に格納されている表を用いて間接的に行なわれる）。

ブロック71、72、……はUMKレジスタが含まれておらず、UAにはもちろん、ブロック70に入っていてキーの全階層の最高レベルを形成する送受用の唯一のUMKだけが

20

存在することがわかるだろう。これら各ブロックは二つの送信キーLMKとLDK、および受信キーの各レベル（この場合、LDK1およびLDK2）に２つのキーを有している。低レベル・キーLDKは比較的稀にしか（例えば50メッセージおきに１回）変らないので、現在のおよび直前のバージョン以外のものを保存しておくことは不必要であり、また高レベル・キーは、たとえ稀でも、変化するので、直前のバージョンの他に現在のものをも保存して丁度それが変化したときに対処しなければならない。

一旦リンクが設定されると、ユーザ・メッセージをUA1からUA2にまたはその逆に送ることができる。リンクは明らかに一つのUAによる要求に応じて設定されなければならないが、一旦設定されてしまえば、それは対象的である。ユーザ・メッセージを送るには、そのプロセスはシステム・メッセージの送出とほとんど同じである。しかし、メッセージ・アセンブリ・レジスタ37のメッセージ本体区画MBは限られた長さしかない。セレクタ・スイッチ76はメッセージ・アセンブリ・レジスタ37のMB区画から暗号化／解読ユニット41への接続経路中に入っており、ユーザ・メッセージに対しては、メッセージの本体は、連続する64ビットのブロックとして、レジスタ部分からではなくPC14から暗号化／解読ユニット41に送り込まれ、暗号化されたメッセージは１ブロックづつPC14に送り返される（PC14はこの点ではインターフェース・ユニット43として動作する）。次にメッセージのMACが計算されてメッセージ・アセンブリ・レジスタ37のMAC区画に送り込まれる。メッセージの長さは、たとえば、MT区画の一部としてあるいはメッセージ本体の最初の部分として含まれている長さ値によって示される。

メッセージ認証コード計算ユニット42は同時に暗号器として動作するように構成することができるので、メッセージ本体の暗号化が始まる前に、メッセージ・アセンブリ・レジスタ37中のMB区画の左側の内容をメッセージ認証コード計算ユニット42へ与え、次にメッセージ本体がユニット41から出て来るにつれて、１ブロックづつそこへ与える。これにより最後のMACがメッセージ本体の最後の暗号化ブロックの直後に利用できる。ただし、MACの計算には実際暗号化と同一のプロセスが含まれているので、実際には暗号化／解読ユニット41を用いて行うのが望ましい。（それ故メッセージ認証コード計算ユニット42は物理的にユニット41とはっきり分れたユニットとしては存在しないが、もちろんその論理的機能は明確に分かれている）。もちろん、この場合には、MACは暗号化と並行して計算することはできず、暗号化の後で計算しなければならない。

ユーザ・メッセージが受信されると、受取りを確認する特別なユーザ・メッセージが自動的に発生され、送信者が要求する場合には、送信元UAに戻される。このような要求は適切なメッセージ・タイプMTで示される。

通信媒体11は信頼性が充分ではないので、通信媒体11

21

によるメッセージ喪失の可能性、二つのメッセージの順序の反転、およびメッセージの重複に対する備えを設ける必要がある。これら設備はユーザ・メッセージとシステム・メッセージとでは異なる。ユーザ・メッセージに対する設備についてここに説明することにする。もちろんメッセージが失なわれたということは、以後のメッセージが受信されるまでは検出することは不可能である。

これらの設備は主として、二つの受信LDKレジスタLDK1とLDK2に関連する1対のビット・レジスタ（ビット・マップ）77と78から構成されている。各ブロック71、72、……はこれらレジスタのそれぞれの組を備えている。ブロック71についての組を第4A図に示す。レジスタ77と78の長さは、ビット数で表わせば、対応する送信元UAのLDKキー・カウンタが0にリセットされるときのカウントに等しい。各ユーザ・メッセージが受信されるにつれて、送信元UAのLDKの使用カウントに対応するビット（これはメッセージ番号KNの一部である）がセットされる。受信されたメッセージに対応するビットが既にセットされている場合には、メッセージを既に受取っていることを示す。したがって今回受取ったバージョンは重複しているものであり、システムによって捨てられる。

ユーザ・メッセージが受信されなければ、通常はシステム動作は起らない。事実、システムは、メッセージ番号が必ずしも連続していないので、喪失されたユーザ・メッセージを識別できるようにはしない。それ故セットされているビットより順番が若いセットされていないビットは、ユーザ・メッセージが未だ受信されていないということではなく、その番号を持つユーザ・メッセージが存在しないということを意味するかもしれない。

システムは、ユーザ・メッセージが脱落していることを、次のメッセージを受取った時点で識別することができるように修正することができる。これは、たとえば、ユーザ・メッセージに、既述のメッセージ番号とともに厳密に連続した番号をも与えることにより、あるいは各ユーザ・メッセージに先行ユーザ・メッセージのメッセージ番号を入れることにより行うことができる。ただし、これを行ったとしても、メッセージが受信されなかったことがわかったときどんな処置を取るかの決定権をユーザの手に残しておくのが望ましい。たとえば見掛け上失なわれたメッセージが無くなったのではなく単に遅れているだけでまだシステムの途中に存在しているということがある。ユーザは事態をそのままにしておくかあるいは彼自身のユーザ・メッセージを失なわれたメッセージの再発信を要求している他のUAのユーザに送るかのいずれかを選択することができる。このような再発信はシステムに関するかぎり全く新しいユーザ・メッセージの送信として行われることになる。新しいメッセージが事実前に送ったが失われたメッセージの繰返しであることの指示を入れるのは送信元ユーザの義務である。

上述のとうり、ユーザ・メッセージが受信されると、

22

確認メッセージの送出が行われる。確認メッセージは特別な種類のユーザ・メッセージとしてシステムによって自動的に発生される。したがって、UAは送られたこのようなメッセージの記録を保存し、この記録は受信の確認が返送されたとき更新されるように構成することができる。これを実現するには、たとえばそのメッセージ・タイプが自動確認であることを示しているメッセージについてのみビットがセットされるビット・マップを用いたり、あるいはこのようなメッセージのメッセージ番号の記録を保存したりすればよい。これが行われると、ユーザは、確認がとられることが必要なそのユーザのユーザ・メッセージのうちのどれがまだ確認されていないかをつきとめ、そのユーザが適当と考えるところにしたがってそれらを再送することができる。もちろん、確認の無いことが必らずしも元のメッセージが意図したデスティネーションに到達していないことを意味するものではない。単にそれに対する確認のメッージが意図するデスティネーションに到達しないことを意味することもある。したがって、ユーザに対して儀礼上の問題および良い慣習として、正しい繰返しであるメッセージを送ったときは必らず、それが前のメッセージの再送であることを示すようにすることが望まれる。

リンクの最初の設定は二つのUAおよびKDCの間の各種の可能なメッセージのシーケンスによって行うことができることが理解されるであろう。このようなシーケンスの二つの例を第IIB表および第IIC表に示す。

第IIB表

UA1→KDC:UA1がKDCにUA2とのリンクを要求する。
UA2←KDC:KDCがUA2にUA1とのリンクを受入れるか尋ねる。
UA2→KDC:UA2が確認し同意する。
UA1およびUA2←KDC:KDCが受信キーをUA1とUA2に送る。
UA1およびUA2→KDC:UA1とUA2が受信を確認する。
UA1およびUA2←KDC:KDCが送信キーをUA1UA2に送る。

第IIC表

UA1→KDC:UA1がKDCにUA2とのリンクを要求する。
UA1およびUA2←KDC:KDCが受信キーをUA1とUA2に送る。
UA1およびUA2→KDC:UA1とUA2が受信を確認し、UA2が受入れる。
UA1およびUA2←KDC:KDCが送信キーをUA1とUA2に送る。

これらのシーケンスは、或る段階で、二つのメッセージが同時にKDCから送出され、且つ二つのメッセージが多かれ少かれ同時にKDCに返送されるという点で、第IIA表のシーケンスより複雑である。また、第IIB表のシーケンスは4段階ではなく6段階から成るので、第IIC表のシーケンスは第IIB表のシーケンスより望ましい。

これら二つのシーケンスにおいて、プロセスはUA2が提案されたリングの受入れを拒絶すれば3番目のメッセージの段階でアポートする。この事態が発生すれば、UA2は拒絶のメッセージを3番目のメセージとしてKDCに送

り、KDCは「リンク拒絶」メッセージを４番目および最終メッセージとしてUA1に送る。最後の二つのシーケンスの場合、各UAはその受信キーを他のUAがその送信キーを受信する前に受信することに注意されたい。これはUAは他のUAがそのメッセージを受信するのに必要なキーを所有するまではこの他のUAにメッセージを送ることができないことを意味する。

第IIA表のシーケンスの場合、UA1はUA2がUA1の送信キーを受取るまではメッセージを送ることができないが、UA2は送信キーをUA1がUA2の送信キーを（UA1の受信キーとして）受信する前にその送信キーを得るので、UA2はUA1がこれを解読するための必要キーを所有する前記にUA1にメッセージを送信することができる。この状況はリンクが最初に設定されるときにのみ発生し得る。そこで、リンクき要求したUA1が最初にメッセージを送りたくなることはありそうなことである。しかしUA1が解読用キーを受取る前にUA2がメッセージを送ろうとすることは起る可能性がある。その結果、メッセージは、メッセージ番号からそれがメッセージを解読するに必要なキーを所持していないことを知ったUA1により拒絶されることになる。ここで一つの選択は単にメッセージを却下して、それが実際上失われるようにすることである。メッセージがシステム・メッセージである場合には、後に説明するようにな処置が取られる。それがユーザ・メッセージである場合には、これは上述のように処理され、この送信はおそらくメッセージが受信されないことを見つけるためのユーザ自身のリソースに委ねられる。あるいはUAはこのようなメッセージを格納してそられを解読するためのキーの受信を持つように構成することができる。

リンクが確立された後、ユーザがリンクの他端のUAとこれ以上通信する必要がないことを確信していれば、またはユーザが他のリンクを設置したいがこのUAが収容できる最大数のリンクを既に持っていてそのため現行のリンクを終結して新しいリンクの余地を作ることだけしかできなければ、このUAはこのリンクを終結させたいかもしれない。これを達成するには、UA1はそれ自身からリンクに関する情報をすべて削除してリンク終結システム・メッセージをKDCに送る。KDCはこれを記録してシステム・メセージをUA2に送りUA2からこのリンクに関するすべての情報を削除することを指示する。KDCは、削除が存在しないリンクに関するものである場合にはエラーとしてリンク削除を記録する。（このような「エラー」はリンクの両端が同時にリンクの終結を要求する場合には自然に発生する可能性がある。というのは、他のメッセージが他方のメッセージのKDCへの到達前にKDCに到達してリンクを終結するからである。）

システム・メッセージ・エラーの回復

上に記したように、メッセージは種々な経緯で「失なわれる」ことがあり、また（通信媒体11のくせによるか

または部外者が記録しては故意に再生することにより）重複することがある。ユーザ・メッセージに関しては、このような事態を処理する方法について上述した。システム・メッセージに関して、このような事態を処理する方法を説明しよう。

システム・メッセージの場合、失なわれるものが皆無で且つ正しい順序で処理されることが肝要である。UAの各リンク（すなわちKDCとの永続リンクおよび他のUAとの各リンク）毎に、そのリンク上に送出される（単なる確認とは別の）システム・メッセージはすべて格納される。これらは以下の二つの状況で記憶装置から除去される。すなわちそれらに対する確認メッセージを受信したとき、またはそれらが冗長になったときである。新しいシステム・メッセージが送出されるたびに、新しいパケットが準備され、このパケットに、記憶装置に入っているすべてのシステム・メッセージが新しいシステム・メッセージをパケットの最後に置いて正しい順序で入れられる。このようにして新しいシステム・メッセージが発生するごとに、未確認でかつ冗長でない古いシステム・メッセージがすべてその前端に付加され、すべてのメッセージ（つまり、古いメッセージプラスこの新しいメッセージ）はパケットとして送られる。それ故受信側では、新しいシステム・メッセージが発生するごとに、すべての未確認システム・メッセージの新しい組合せを正しい順序で受信する。そこで、そのパケット中のどのメッセージの前にもすべての未確認かつ非冗長メッセージが正しい順序で並んでいるので、受信側は必然的にシステム・メッセージを正しい順序で必ず処理することになる。もちろん、受信側はその時点までにこれらのシステム・メッセージのうちのあるものを含んだより以前のパケットを受取ったことがありそのシステム・メッセージについては既に処理がなされていたかもしれない。受信側は、リンクごとに、処理を行なった最後のメッセージの記録（メッセージ番号による）を保存しているので、重複しているメッセージ、特に今受取ったパケットに入っているそのようなすべてのメッセージを含めて、すべて無視する。受信側は新しいパケットが届くとすぐにそのパケットに入っているメッセージへの応答を開始する。

確認メッセージはメッセージの受信を確認する以外の何者でもない単なる確認メッセージであることがあり、あるいは或る情報を運ぶ普通のシステム・メッセージであることもある。後者は着信システム・メッセージに応答して発せられるので、その先行メッセージを暗黙裡に確認する。システム・メッセージはその効果が後のものにより取消されると冗長なものになる。たとえば、リンクの設定を要求するメッセージはそのリンクの解消を要求する後のメッセージにより取消される。

厳密に言えば、重複メッセージは完全に無視されるのではない。重複メッセージが検出されると、単なる確認

25

が送信側に送り返されるが、このメッセージにはそれ以上の処理が加えられることはない。これは通常のメッセージの確認がシステム内で失なわれてしまっていることがあるからである。仮に送信側がそのメッセージに対する確認を以前に受取っていれば、そのメッセージは再送されなかったであろう。そこで、もし重複メッセージの確認が送られなければ、送信側はそれを繰返して送り続けるであろう。だが、送信側がメッセージに対する確認を受取れば、確認メッセージのメッセージ番号以上のメッセージ番号を持つすべてのメッセージを再送記憶装置から安全に削除することができる。何故なら、メッセージは、すべての先行メッセージが順当に受信されている場合に限り、受信側によって受信され、処理され、且つ確認されることができるからである。

これにより、すべてのシステム・メッセージが正しい順序で確実に処理を受けることが保障され、また新しいメッセージが発生されるごとに行われる自動再送信により最後のメッセージがより以前のメッセージの再送によって遅れることがないということが保障される。その他に、メッセージを再送する第２の方法がある。メッセージ・パケットの送信後新しいメッセージを発生せずかつ確認も受取らないままに充分長い時間が過ぎたら、記憶装置95に記憶されている現メッセージのパケットが自動的に再送される。

このようなパケットを構成することにより、メッセージは常に正確に同じ形で送信される。ただし、メッセージは現在の基本キーのもとで再暗号化される。パッケージ全体をひとつのユニットとしてまたは単一のメッセージとして送信することが可能である。ただし、その中の個々のメッセージが解読されるにつれて処理を受けることができるような形で送ることが望ましい。そうすれば、メッセージが中断されたり傷つけられたりした場合、その一部分だけが失われ、受信側は最新の状態に向かって途中まで進んだ状態に居ることができるからである。これが意味しているのは、パッケージの認証は、ここの事態は発生する可能性はあるがパケットはなお妨害から保護されていて部外者がシステムを偽のメッセージに応答するようにだますことができないように設計しなければならないということである。

パケットのフォーマットは、メッセージ・アセンブリ・レジスタの左端（SC、DN、MT、KN、およびMKの各区画）にある通常の「ヘッダ」情報で始まる。MT区画の内容はメッセージが二つ以上のメッセージのパケットであることを示すインジケータ・ビットを含んでいる。KN区画のメッセージ番号は今のメッセージのメッセージ番号である（このメッセージはパケットの最後にあるもの）。パケットの最初のメッセージは格納されているメッセージである。このため、パケットに入っているすべての格納メッセージに関しては、ソースコードやデスティネーションコードは必要がなく、特別なMKも必要がない。したがってそれは省略形メッセージとして組立てられ、そのメッセージタイプ（もしこれが一連の格納メッセージの最後のものでない場合にはインジケータ・ビットが付いている）、メッセージ番号KM、およびメッセージ本体MB（もしあれば）から構成される。またPMAC区画を備えており、これは（単一メッセージに関して）空白である。このように組立られたパケットのMACが計算されてMAC区画に入れられる。

26

パケットの次の区画を今度は、次の格納メッセージを、あるいは格納メッセージがもうすべて入れられてしまった場合には、現行メッセージを、取入れて組立てる。このため、パケットの前のメッセージに対して計算したばかりのMACをPMAC（前のMAC）部分に入れ、このPMAC値を暗号化し、パケットの現在の部分に取入れられているメッセージについて計算された新しいMACによりカバーされているフィールドに入れる。このプロセスは現メッセージがパケットの最終部分を形成するまで続けられる。現メッセージのMB区画にはMT区画およびKN区画は含まれない。なぜならこれらは既にパケットのヘッダに入っているからである。）

このパケット構造ではパケットを１メッセージづつ解体し、解読し、処理することができることが明らかである。その上、個々のメッセージおよびそのシーケンスはともにMACのシーケンス鎖によって認証される。各MACはそれに先行するメッセージの完全性を確認し、各メッセージはそれに入っている前のメッセージのMACを所持しているので、シーケンスの変化（メッセージの順序変え、削除、または繰返し）があれば、シーケンスを逸脱したメッセージが届くとすぐにそのMACはチェックを通らなくなることになる。

受信側は、パケット中の個々のメッセージに個別に応答する。ただし、応答メッセージはどれも（単なる確認メッセージ以外の）即座には送出されずにメセージ記憶装置に入れられ、その着信パケットが完全に処理されてしまってはじめて、これらの応答メッセージは単一パケットとして（古い未確認メッセージとともに）送出される。（もしこうしないならば、これら応答は、一連のより長いパケットとして、繰返し送出しなければならなくなる。）

上記のように、パケットの長さは、メッセージを鎖状に接続すること、および各メッセージのMTに、以後に続くメッセージが存在するか否かを示すビットを入れることにより黙示的に示される。これによりもちろん受信側は、MB本体にそのMTとKNが入っている再送信メッセージと、パケットのヘッダにそのMTとKNが入っている現メッセージ（パケットの最後のメッセージ）とを区別することができる。代りの技法は、ヘッダ中に、たとえばMT区画またはKN区画の一部としてパケット長さ値（メッセージの数）を入れることである。

新しいシステム・メッセージが発生するごとに未確認

システム・メッセージのすべてをこのように再送信する方式では、不必要な再送信は非常にわずかしか起らない。再送信が不必要であると正当に言うことができるのは、メッセージは正しく受信されたがその確認を元のUAが未だ受取っていない場合か、あるいはその確認が道に迷ってしまった場合だけである。代案といては、受信側が受取った最後のメッセージのメッセージ番号の記録をとっておき、新しいメッセージがその番号の次の順番のメッセージ番号を持っていないことがわかった場合に、失われたメッセージの再送信の要求を送ることである。しかしこの技法は厳密に連続したメッセージ番号が使用することを必要としており、また受信側が現メッセージを処理することができる前に二つのメッセージ伝達（要求と応答）をするという遅れを生じ、これら「回復」メッセージが道に迷った場合には、なお更に遅れる。システム・メッセージは比較的短く、それ故１つのパケットにより未確認メッセージを皆再送信する費用は高価になりそうもないということも注目しておいてよい。これはユーザ・メッセージの場合とは対照的である。ユーザ・メッセージの長さは非常にばらつきやすく、且つ非常に長いことがあるからである。

パケットは（単一のユーザ・メッセージと比較して）かなりのまた可変の長さを持っているから、パケットは再送信のためユーザ・メッセージと大まかには同じ方法で準備され、連続するブロックは暗号化、解読ユニットに送り込まれ、暗号され認証されたメッセージは、発生されるに従ってインターフェース・ユニット43として働くPC14に蓄積される。

これらの動作に関係する装置を第３図、第４図、および第５図に示す。端末（UAまたはKDC）には、その端末からの各リンク毎にそれぞれのシステム・メッセージ格納ブロックがある。すなわち、KDC内にはすべての端末UA1、UA2、UA3、……へのリンクについてブロック85、86、87、……（第３図）があり、また各UA端末にはKDCへのリンクとリンクされている端末UA－Ｉ、UA－II、……とに関するブロック90、91、92、……（第４図）がある。これらブロックはもちろんメッセージアセンブリ処理回路51または75にマルチプレクサ60または74を介して接続されている。第５図はブロック85の主要構成要素を示す。他のブロックは実質的に同じである。未確認システム・メッセージを格納する記憶装置95があり、これは幾分FIFO（先入れ先出し）記憶装置のように動作するが、非破壊読出しが行なわれる。システム・メッセージは上からこの記憶装置95に送り込まれ、それらが削除されるまで着実に下に移っていく。記憶装置95の中のメッセージにはそれと対応してそのメッセージ番号KNが区画96に格納されている。レジスタ97は最後に確認されたメッセージのメッセージ番号RXKNを格納しており、これが変ると記憶装置95中のメッセージは、メッセージ番号RXKNと一致するメッセージ番号を区画96中に有しているメッセージまで上向きに削除される。新しいシステム・メッセージが用意されつつあるときは、なお記憶装置95に入っている古いメッセージはすべて上向きに、すなわち最も古いものが最初に、非破壊的に読出される。次に新しいメッセージが記憶装置95の最上部に入る（また既に記憶装置95に入っているメッセージはすべて押し下げられる）。

事実には、キー階層内の基本キーのレベルによって、二つのクラスのシステム・メッセージがあり、それらはシーケンスの異なるメッセージ番号KNを有している。したがって記憶装置95とレジスタ97は二重になっているので、二つのクラスのメッセージは別々に格納され、ブロック85は二つのクラスのための二つの区画ＡおよびＢから構成されている。高位の基本キー（すなわち階層の高い基本キー）のメッセージはすべてパケット内で低位の基本キーのメッセージに先行する。このことはメッセージは元々発生した順序と厳密に同じシーケンスでは送られないということを意味する。ただし、この影響は幾つかの新しいキーがそうでない場合よりわずかに早く到達することがあるということだけである。

ブロック85はまた、システム・メッセージが最後に送り出されてから経過した時間を測定するのに使用されるタイマTMR98を有し、このタイマ98の時間が予め設定された限界を超過したときまだ確認されていないシステム・メッセージの再送信をトリガする。このタイマ98はパケットが送り出されるごとに０にリセットされる。

各UAのブロック90、91、92、……は安全モジュールに入っていて、再送信を待っているメッセージのリストを考えられ得る部外者から安全に守るようになっている。ただし、KDCでは、対応するブロック85、86、87、……は安全モジュールには入っていなくて、色々な理由のため、支援用記憶装置に入っている。KDCにはすべてのUAとのリンクがあるので、格納されているメッセージの数はUAのものよりはるかに多いと思われる。格納メッセージの喪失（たとえば計算ユニットコンピュータ）18の故障による）は、（後に説明するように）KDCはバックアップおよび復元の手続を所持しており、またKDCはUAより部外者による攻撃が少いと思われるので、UAでの対応する喪失ほど重大ではない。支援用記憶装置に格納されているこのKDC情報は、偶然のまたは故意の変造に対して、次に説明するローカル・メッセージ格納技法によって保護されている。

ローカル・メッセージ記憶装置

UAにメッセージを安全に格納できることが望ましい状況が存在する。したがってユーザは、ユーザ・メッセージが受信されたときそこにいないかあるいはそのメッセージを保存しておきたいかのいずれかのため、受信したユーザ・メッセージを安全に格納しておきたいことがある。またユーザは、UA内に、自分が発生したユーザ・メッセージのような資料を安全に格納したいことがある。

本システムはこれら両方の設備を提供する。

受信したメッセージをUAに格納する場合には、受信したときの形、すなわち解読してない形でディスクメモリ15等の支援用記憶装置に格納する。このことは、部外者が格納メッセージにアクセスすることができたとしても、通信媒体11に載っているメッセージを傍受して得ることができた以上の知識を得ることができないこと、特に、通信媒体11に現われたままのメッセージを支援記憶装置に格納されているメッセージと比較しても何も得るところがないことを意味する。ただしユーザはもちろん自分自身で後にメッセージを解読することができなければならない。したがって、メッセージにはそれを暗号化したLDKが付属している。このLDKは、キーが安全モジュールの外側に平文で存在することを許容され得る状況はないので、それ自身暗号化された形になっていなければならない。それでこれはキー階層でその上にあるLMKのもとで暗号化されて格納されている。そのLMKも、再び暗号化された形の、階層の最上部にあるLMKのもとで暗号化された、メッセージに付属している。LMKそれ自身は、階層の最上部にあるので、暗号化することができず、メッセージの一部として平文で格納することもできない。その代り、メッセージが格納される時にこの識別番号を付加する。

キーは二つの異なる形で現われるべきではないということが重要である。各キー（UMKは別）基本キー（その上位のキー）およびメッセージ・キーのもとで暗号化されて受信された。キーは安全モジュールのブロック70、71、72、……に平文で（すなわち解読されてから）格納される。各キーはそのため受信されたときの暗号化されが形で、その暗号化に使用されたMKとともにこれらのブロックに格納される。キーがメッセージに付加されると、格納されている暗号化された形態および関連するMKが付属部を形成するのに使用される。

UAはUMK履歴記憶装置UMKH105（第4図）を備えており、これには現在のおよび過去のUMKがそのシリアル番号（識別番号）とともに格納されている。新しいUMKがKDCブロック70に入ると、それはUMKH履歴記憶装置105にも入る。メッセージへの一連の付属部を発生するには、各種レベルのキーを今度はメッセージ・アセンブリ処理回路75の中で、それぞれその上位のキーのもとで暗号化し、最後に現UMKのシリアル番号をブロック70のUMKキー番号レジスタ40A（第2図）から取る。（UMK履歴記憶装置105の容量は有限であるから、一杯になれば、過去の古いUMKがそれから取出されて、もっと最近のUMKのもとで暗号化された上でディスクメモリ15に格納される。）

格納されたメセージを回復するには、付属部を一つづつ第4図の回路に送る。この際、最初のものはUMK履歴記憶装置105から適切なLMKを得るのに使用するUMKのシリアル番号であり、次の付属部はUMKのもとで解読してLMKを得るためにメッセージ・アセンブリ処理回路75に送

られ、最後の付属部はこのLMKのもとで解読してLDKを得るためメッセージ・アセンブリ処理回路75に同様に送られ、次にメッセージ自身がLDKおよびメッセージに埋込まれているMKのもとで解読してメッセージの本体を得るためメッセージ・アセンブリ処理回路75に送られる。

実質的に同じ技法がローカルに発生されたメッセージを安全に格納するに使用される。安全格納キーブロSSKL07は、ブロック70、71、72、……と同様であるが、ローカル・キー階層PMK、PSMK、およびPDKのための一組のキー・レジスタを備えている。（安全格納キーブロック106は、「送信」キーに対応するキーだけしか備えていないので同様なブロックより小さく示してある。明らかに、「受信」キーに対応するキーを格納する必要性はない。）メッセージを格納するには、メッセージを、メッセージ・キーKMと現行のPDKとを用いて通常の方法で暗号化する。これにより、メッセージには、PSMKのもとで暗号化されたPDK、PMKのもとで暗号化されたPSMK、現行UMKのもとで暗号化されたPMK、および現行UMKのシリアル番号が皆格納のため付加される。メッセージは他のUAから受取られ安全に格納されたメッセージに関して行なうのと実質的に同様にして解読することにより回復することができる。

システムはまた、他のUAから受信したものであろうとローカルに発生したものであろうと、ローカルに格納されているメッセージの認証を行う。このような認証の目的はローカルに格納されているメッセージを、もちろん安全モジュール16ではないがPC14やディスクメモリ15等の記憶装置（第1図）にアクセスすることができる部外者による妨害から防護することである。このような部外者はメッセージを削除し、メッセージを変更し、あるいはメッセージを挿入しようとするかもしれないからである。

ディスクメモリ15等の記憶装置（またはPC14の内部記憶装置）には長さがいろいろで且つ記憶装置全体に渡っていろいろなロケーションに配置された各種のメッセージ111（第6図）が入っている。（もちろん個々のメッセージは、ここに述べた原理に影響を与えることなく、連続しないページの系列に普通の仕方で配置することができる。）これらメッセージと関連してディレクトリ112がある。このディレクトリ112は区画113に各メッセージの識別用タイトルとディスクメモリ15の中のメッセージのロケーションをリストするものである。メッセージがディスクメモリ15に入ると、それに対応するMACがメッセージ認証コード計算ユニット42により、メッセージ自身がそのもとで暗号化された基本キーを用いて、計算される。このMACはディレクトリ112の区画114に、区画113に格納されているメッセージのタイトルとロケーションに関連付けて格納される。その他に、ディレクトリ112のMACの全体のリストが特殊メッセージとして取扱われ、これらMACに対してグローバルMACすなわちスーパー

31

MACが計算される。このグローバルMACは安全モジュール16の内部に設置されたグローバルMACレジスタ115に格納される。

ディレクトリ112にリストされているところの格納されているファイルの個々の完全性をチェックしたい場合には、そのMACを計算し、ディレクトリ112に格納されているMACと比較する。これらMACは暗号化キーを用いて計算されるので、部外者がファイルを修正しようとしても、修正したファイルの正しいMACを作ることができない。したがってディレクトリ112のMACはその個々のファイルを認証する。ファイルの全セットの完全性をチェックしなければならない場合には、ディレクトリのMACのグローバルMACを計算し、安全モジュール16の中のMACコンパレータ44（第2図）により、グローバルMACレジスタ115に格納されているグローバルMACと比較する。部外者がディレクトリ112を何らからの仕方で、たとえばエントリを削除したり、エントリの順序を変更し、あるいはエントリを挿入したりして、変更すれば、グローバルMACが変ることになる。そしてグローバルMACは安全モジュール16に格納されていてこれには部外者がアクセスすることができないから、弐外者はそれを変更することができず、変造されたディレクトリのグローバルMACはグローバルMACレジスタ115に格納されているグローバルMACと合わないことになる。（MACはすべてキーを用いて計算されているので、部外者は変更されたファイルのグローバルMACを計算することはできない。しかし仮にグローバルMACにアクセス可能であったとすれば、部外者は前のバージョンによりファイル全体およびグローバルMACを交換することができる。）

もちろん、個々のファイルのMACを格納されたファイルの一部として格納することができることは理解されるであろう。この場合、グローバルMACの計算にあたってはディレクトリ112を使用して、そのメッセージから格納されている各MACが探し出される。また、ディレクトリ112が充分大きければ、これを区画に分割して、区画MACをその区画で識別されるメッセージのMACから各区画について計算し、グローバルMACを区画MACから計算するようにすることができる。区画MACは平文で格納することができる。この場合これら区画MACは部外者が修正することができるが、そのような区画MACはその区画に関連するメッセージから計算したMACとうまく合致しないか、あるいはそのグローバルMACがグローバルMACレジスタ115に格納されているグローバルMACとうまく合致しないことになる。ディレクトリ112はもちろんディスクメモリ15に設置することができる。グローバルMACを安全モジュール16のレジスタに格納するかわりに、これを安全モジュール16の外部に格納することができる。ただし、これは格納された情報のすべてを以前のバージョンで検出されることなく置換えることができるという上に記した危険を冒すことになる。

32

ユーザが格納されているメッセージを、たとえばメッセージを変更し、新しいメッセージを追加し、あるいはメッセージを削除して、変更したい場合には、付与された、すなわち加えられたメッセージのMACを計算してディレクトリ112に格納するか、あるいはディレクトリ112から削除されたメッセージのMACおよび新しいグローバルMACを計算してグローバルMACレジスタ115に格納するかしなければならない。これには新しいMACの計算（これはメッセージの内部の認証に必要である）とメッセージMACからの新しいグローバルMACの計算が行なわれるだけである。変更されないメッセージのMACは不変であり、これらメッセージの処理は不要である。

UAの変更

ユーザがUAを彼自身のUAであるUA1から別のUAであるUA2に一時的にまたは永久的に変えたいことがある。一時的に変えたい場合は、ユーザは自分の通常のUAに向けられたメッセージを読むのに一時的に新しいUAを使用することができるようにしたくなる。また永久的に変えたい場合は、ユーザは自分の古いUAから新しいUAに全てを転送したくなる。これら二つの場合の取扱いは異なる。

前者の場合では、ユーザはKDCに、他のどの端末を使用したいかを指定して旅行キー（journey key）を要求する。KDCはこれを受けると直ぐユーザに旅行キーを発行し、ユーザが訪問して旅行キーに応答するUAを設定し、旅行キー（UM2のUMKおよびCDKのキー階層のもとで暗号化されている）をUA2に送り、ここでUA1のアドレス・コードとともに旅行キー・レジスタ107に格納される。ユーザはまた受信したすべてのメッセージを格納するとともにUA2からの呼出しに対してそれらをUA2に送って応答するため、彼自身のUAを設定する。このメッセージの転送はUA1がメッセージを解読し、これを再び旅行キー（通常のランダムなMKとともに）のもとで暗号化してから、修正したメッセージをUA2に送ることにより行われる。UA2では、ユーザはメッセージを解読するのに彼の旅行キーを使用する。この技法では同じメッセージを相異なる複数キーのもとで暗号化して送信するということがあり、また所与の使用後は旅行キーを更新できないので、利用に当っては注意しなければならない。また、後者の場合では、ユーザのUMKをUA2に物理的に輸送し、そこに設置しなければない。（実際、以前の全てのUMKも同様に設置して、安全に格納されているメッセージを転送することができる。）次にKDCとのリンクを上述のように確立し、次に他のUAとのリンクを確立する。UA2に既に格納されてたキーはすべて、もちろん、新しいユーザのUMKが設置される前に破壊され、UA1の中のキーもすべて同様に破壊される。UA1に安全に格納されているすべてのキーは、更に暗号化されることなく、すなわち暗号化されたメッセージの格納形態プラスUMKのシリアル番号までの付属部という形で、UA2に送られるので、新しく設置されたUMKのもとでUA2において解読する

33

ことができる。

KDCメッセージの記録

UAでは、キーに対する、すなわち安全モジュールの内容に対するバックアップ・システムが存在しない。これは安全モジュール外で利用できるキーを設けることは重大な弱点となるからである。UAで起きた障害はUAを再起動させることによってのみ回復できる。KDCは各UAに対するUMKの完全なセットを保持しており、適切なセットをキー分配径路13を通して送り且つ設置することができるので、安全に格納されているメセージをすべて読取ることができる。そこでUAを再設して安全に格納されたメッセージを読取ることができるようにしなければならない。次にUAはまずKDCとのリンクを、次に接続したい他のUAとのリンクを再び確立しなければならない。（UAリンクの最初の設定の場合のように、この動作のうちの多くの部分はキー分配径路13を通ってKDCから伝えられた一組の格納メッセージにより行うことができる。）その障害期間中それに向けられたメッセージはすべて失なわれており取出し不能になる。自分のメッセージが失なわれてしまったユーザは、故障したUAが回復し、そのリンクが再確立された時点で、そのメッセージを再送したいか否かを決定する責任がある。

KDCの故障を処理する設備はこれとは異なる。KDCは自分が送受したすべてのメッセージの記録つまりログを、それが処理を受けた順序に、保持している。このログは支援用の記憶手段19に保持されている。また、KDCの状態が記憶手段19に定期的に格納される。KDCに故障が発生すると、オペレータはKDCを以前に格納された状態まで記憶手段19から回復しながらバックアップしなければならない。次にそのとき以後に発生したすべてのメッセージのログをKDCに再生して戻す。これによりKDCがその正しい現在の状態にまでなる。ただし、その時間中にKDCが発生し送出したキーはすべて失なわれている。したがって、ログの再生中、キーの発生および送出に関係しているメッセージは反復され、したがって新しいキーがUAに送出されて、先に送出されたがKDCでは失なわれたものと置き換わる。このようにしてシステム全体が一貫した状態に回復する。

〔発明の効果〕

以上詳細に説明したように、本発明によれば、通常部分的な更新しかないシステムで認証コードの再計算が大幅に簡単になる。

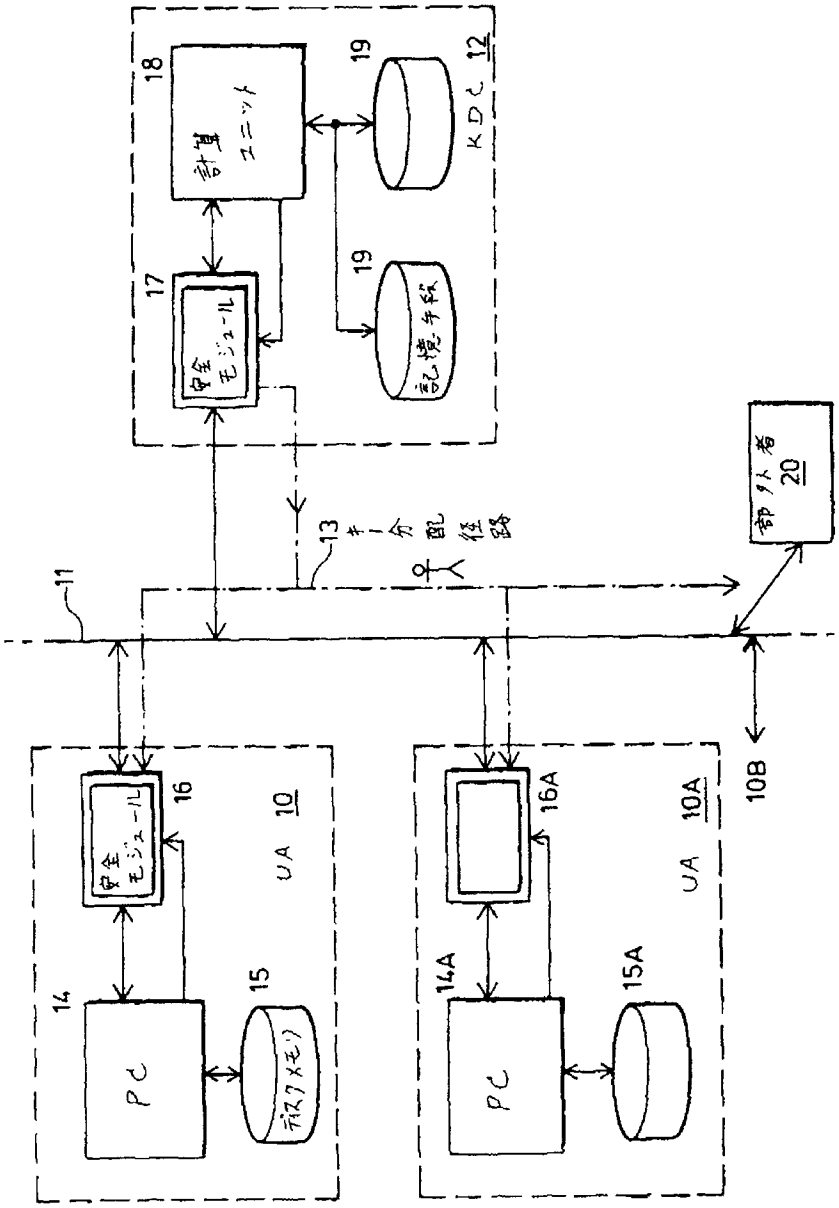
【図面の簡単な説明】

第1図は本発明の一実施例の全体的構成を説明するための図、第2図は第1図中の端末の主要部の構成を説明するための図、第3図は第1図中のKDCの主要部の構成を説明するための図、第4図は第1図中の端末の他の主要部の構成を説明するための図、第4A図は第4図の部分的構成を示す図、第5図は第3図中のKDCにおける再送動作を説明するための図、第6図は第1図中の端末の他の主要部の構成を説明するための図である。

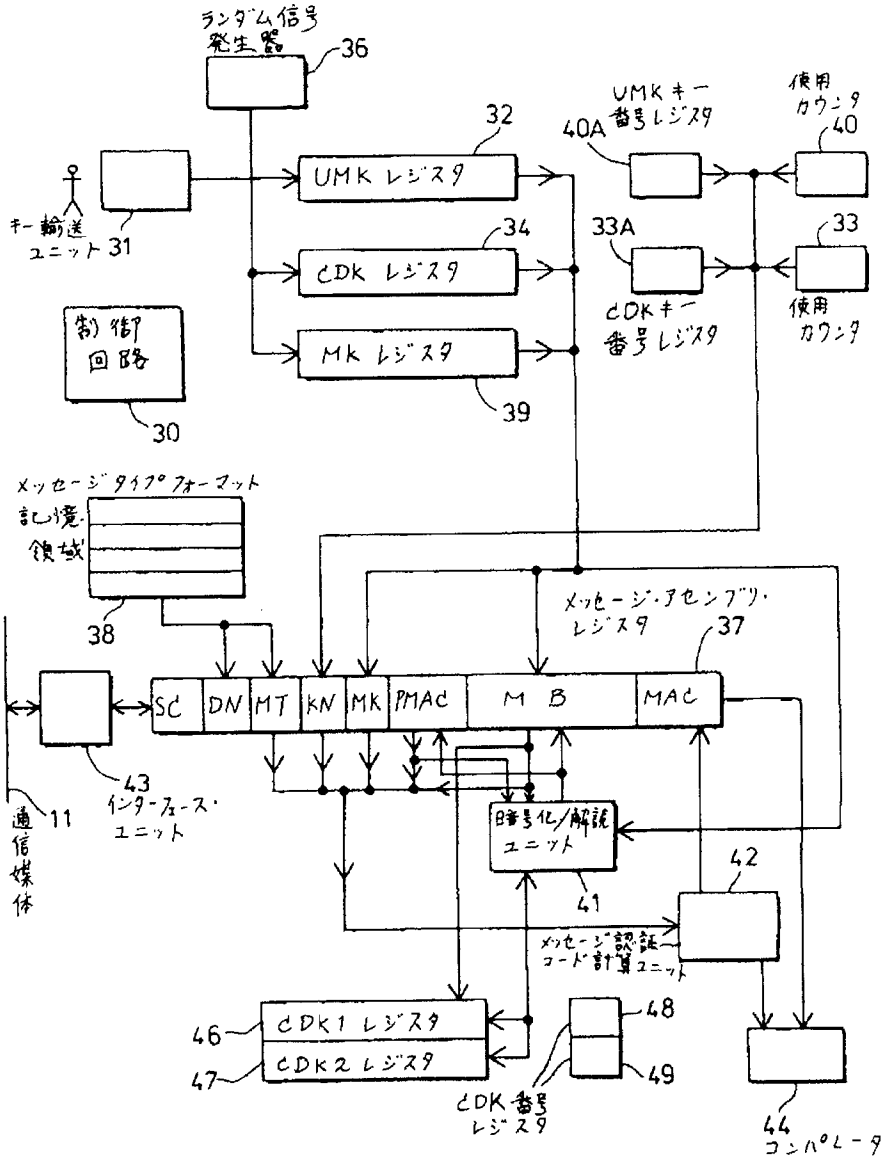
34

10,10A,10B:端末
11:通信媒体
12:KDC
13:キー分配径路
14,14A:PC
15,15A:ディスクメモリ
16,16A,17:安全モジュール
18:計算ユニット
19:記憶手段
20:部外者
30:制御回路
31:キー輸送ユニット
32:UMKレジスタ
33,40:使用カウンタ
33A:CDKキー番号レジスタ
34:CDKレジスタ
36:ランダム信号発生器
37:メッセージ・アセンブリ・レジスタ
38:メッセージタイプフォーマット記憶領域
39:MKレジスタ
40A:UMKキー番号レジスタ
41:暗号化／解読ユニット
42:メッセージ認証コード計算ユニット
43:インタフェース・ユニット
44:コンパレータ
46:CDK1レジスタ
47:CDK2レジスタ
48,49:CDK番号レジスタ
50:制御ユニット
51:メッセージ・アセンブリ処理回路
52:メッセージ・アセンブリ・レジスタ
60:マルチプレクサ
61:セレクタ回路
73:レジスタ
74:マルチプレクサ
75:メッセージ・アセンブリ処理回路
76:セレクタスイッチ
77,78:ビット・レジスタ
95:記憶装置
97:レジスタ
98:タイマ
105:UMK履歴記憶装置
106:安全格納キー・ブロック
107:旅行キー・レジスタ
111:メッセージ
112:ディレクトリ
115:グローバルMACレジスタ

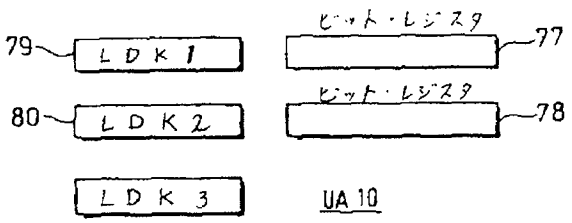
【第1図】


18
計算ユニット
19
KDC 12
17
安全モジュール
19
記憶手段
13
キー分配経路
11
外部者 20
10B
16
安全モジュール
UA 10
14
PC
15
ディスク
16A
UA 10A
14A
PC
15A

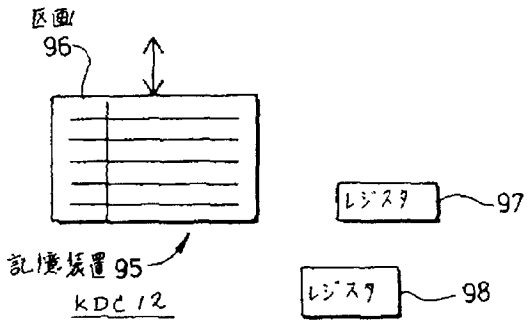
【第2図】


ランダム信号発生器
36
32
UMKレジスタ
UMKキー番号レジスタ
40A
使用カウンタ
40
キー輸送ユニット
31
34
CDKレジスタ
33A
33
CDKキー番号レジスタ
使用カウンタ
制御回路
30
Mkレジスタ
39
メッセージタイプ°フォーマット記憶領域
38
メッセージ・アセンブリ・レジスタ
37
SC
DN
MT
kN
MK
PMAC
M B
MAC
43
インターフェース・ユニット
11
通信媒体
暗号化/解読ユニット
41
42
メッセージ認証コード計算ユニット
CDK1レジスタ
CDK2レジスタ
46
47
48
49
CDK番号レジスタ
44
コンパレータ

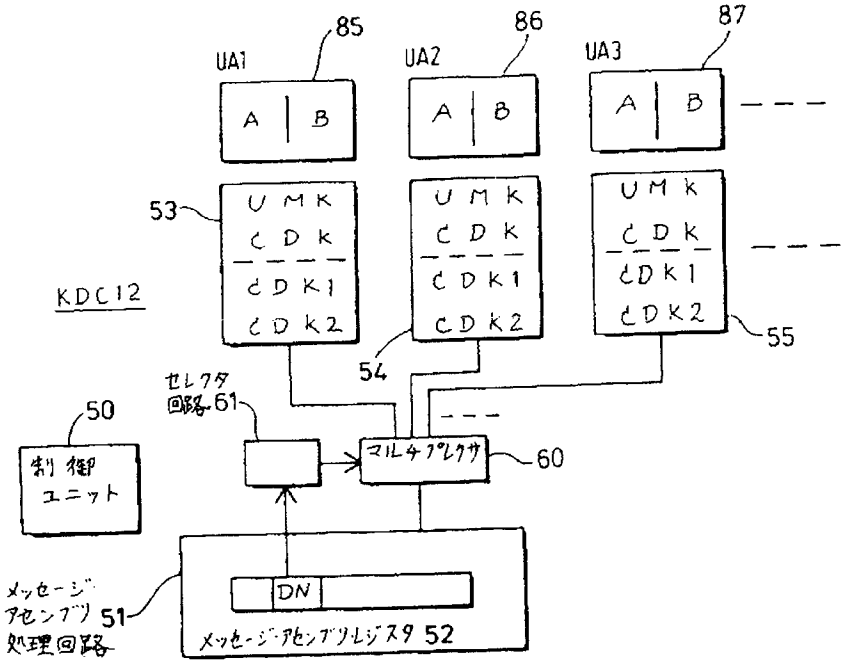
【第4A図】


79
LDK1
ビット・レジスタ
77
80
LDK2
ビット・レジスタ
78
LDK3
UA 10

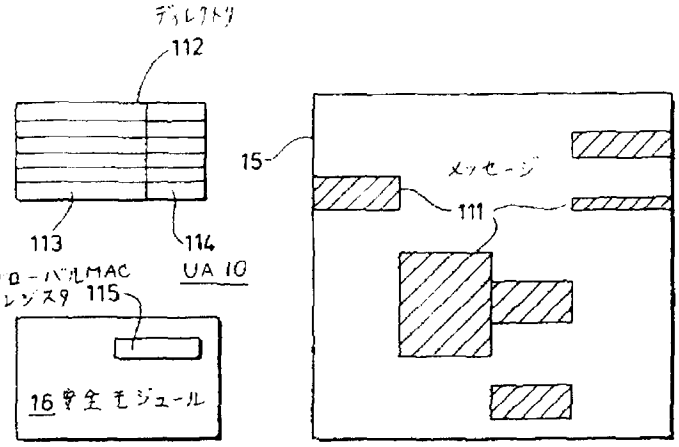
【第5図】


区画
96
記憶装置 95
KDC 12
レジスタ
97
レジスタ
98

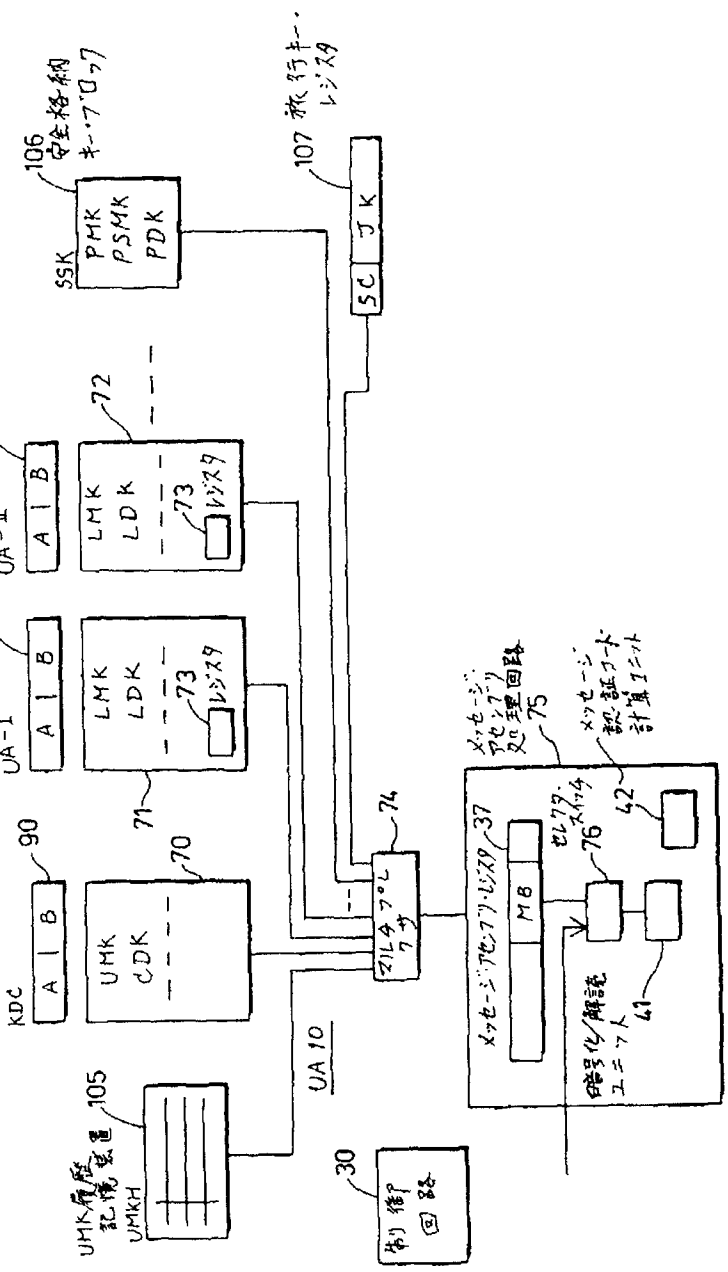
【第３図】


UA1
85
UA2
86
UA3
87
A
B
UMK
CDK
CDK1
CDK2
53
54
55
KDC12
セレクタ回路 61
50
制御ユニット
マルチプレクサ
60
メッセージアセンブリ処理回路
51
DN
メッセージアセンブリレジスタ 52

【第６図】


ディレクトリ
112
113
114
UA 10
グローバルMACレジスタ 115
16 安全モジュール
15
メッセージ
111

【第４図】



フロントページの続き

(72)発明者　グレーム・ジエイ・プラウドラ
イギリス国イングランド・ビーエス12・6エクスキユー・ブリストルストーク・ギフオード・ミード・パーク・タツチストーン・アベニユー5

(56)参考文献　特開　昭60－26387（ＪＰ，Ａ）
特開　昭61－275940（ＪＰ，Ａ）
情報処理　Ｖｏｌ．25　Ｎｏ．６
ｐ．566－574